

Impact Case Study: How long does your real-time software take to run?

Robert I. Davis
Real-Time Systems Research Group,
Department of Computer Science,
University of York, York, UK.
rob.davis@york.ac.uk

Guillem Bernat, Ian Broster, Antoine Colin
Rapita Systems Ltd.
York, UK.
{guillem.bernat, ian.broster, antoine.colin}@rapitasystems.com

Impact Summary

Research from the Real-Time Systems Group at the University of York resulted in an innovative Worst-Case Execution time (WCET) analysis technology now called RapiTime, which was transferred to industry via a spin-out company, Rapita Systems Ltd. The technology enables companies in the aerospace, space and automotive industries to reduce the time and cost required to obtain confidence in the timing correctness of the systems they develop. The RapiTime technology has global reach having been deployed on major aerospace and automotive projects in the UK, Europe, Brazil, India, China, and the USA. Key customers include leading aerospace companies as well as major automotive suppliers.

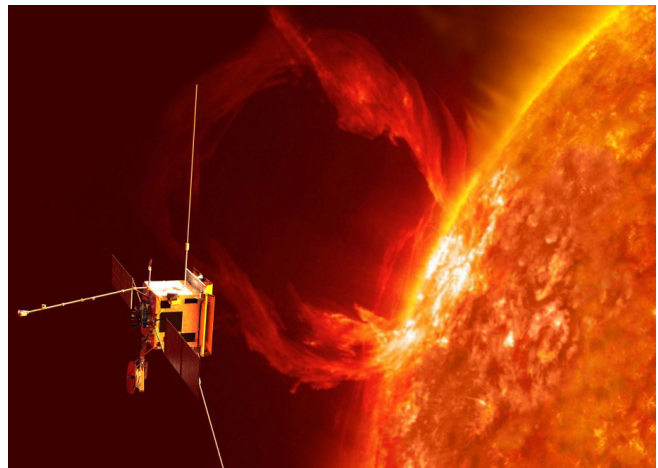


Background

Determining the longest time that software components can execute on a microprocessor, referred to as the Worst-Case Execution Time (WCET), is a key issue in the development of real-time embedded systems in the aerospace and automotive industries. Here, intermittent timing failures caused by software exceeding its budgeted execution time can lead to operational problems, reliability issues, and in some cases catastrophic consequences. In these applications the WCET of software components needs

to be tightly bounded to avoid the need to overprovision hardware in terms of faster, but more costly processors.

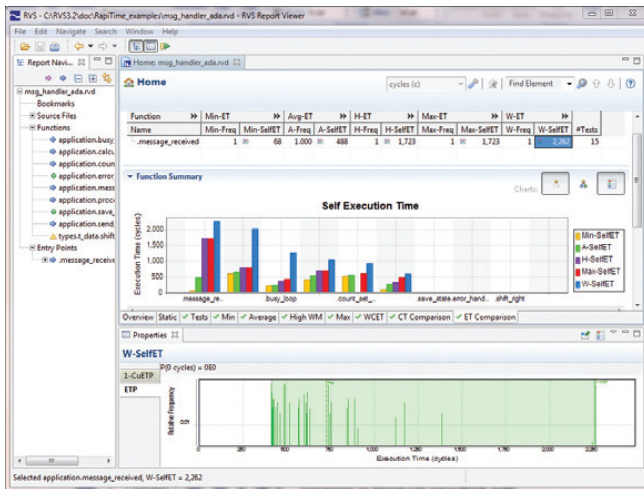
Prior to this research, there were two main approaches to WCET estimation; end-to-end measurement and static analysis. End-to-end measurement techniques insert profiling code into the software. During testing this profiling code records the end-to-end execution time of each invocation of each software component. End-to-end measurement alone typically under-estimates the WCET, and provides little confidence that timing constraints will always be met during operation. Static analysis techniques analyse the software object code and compute the WCET using a model of the timing behaviour of the microprocessor. This is done without running the code. Using static analysis alone has the disadvantage that the computed WCETs depend on the model of the processor and its hardware acceleration features; as processor technology advances this becomes more complex, expensive and in many cases is infeasible to do.



Research

During the NextTTA project (2002 to 2004) four members of the Real-Time Systems Research Group (RTSRG) at the University of York, Guillem Bernat, Antoine Colin, Stefan Petters, and Alan Burns developed a set of hybrid and probabilistic techniques for WCET

analysis [1], [2], [3], [4], and [5], now referred to as RapiTime. The RapiTime approach combines static analysis of the structure of the source code with timing measurements taken during testing, which record the execution time of short sub-paths through the code. RapiTime recognises that the best possible model of an advanced microprocessor is the microprocessor itself and therefore uses *online* testing to measure the execution time of short sub-paths in the code. By contrast, *offline* static analysis is the best way to determine the overall structure of the code and the paths through it. Therefore RapiTime uses path analysis techniques to build up a precise model of the overall code structure and determine which combinations of sub-paths form complete and feasible paths through the code. Finally the measurement and path analysis information are combined using mathematical techniques to compute WCETs in a way that captures accurately the execution time variation on individual paths due to hardware effects.



This novel and innovative approach combines the advantages of both measurement and static analysis techniques while avoiding their drawbacks. Unlike static analysis, it does not require the expensive and time consuming production of a precise timing model for each new microprocessor variant and its hardware acceleration features, and so is portable to a wide range of different microprocessors. RapiTime is also viable when the only accurate timing model that is available is the microprocessor itself. Further, RapiTime does not require the plethora of manual annotations that static analysis alone needs to establish essential information about control flow. This greatly reduces the amount of engineering time required before meaningful results can be obtained, and removes a potential source of errors. Compared to measurement, RapiTime is able to identify the worst-case path and compute the overall WCET of software components from the WCETs of sub-paths when not all of the complete paths through the code have been executed. This significantly reduces the amount of testing required to verify timing correctness.

Route to Impact

During the EU FP5 NextTTA project members of the RTSRG group, Guillem Bernat, Antoine Colin, Stefan Petters, and Alan Burns, introduced research on hybrid measurement-based WCET analysis. This approach combined both measurement and static analysis techniques to accurately estimate the WCET of complex software components running on advanced microprocessors. As part of the project, they also developed a prototype WCET analysis tool called pWCET [5]. This tool was evaluated on an Audi drive-by-wire system. Audi was an industrial partner in the NextTTA project. Audi's expression of interest in pWCET and its capabilities led directly to the formation of a spin-out company to transfer this technology into industry.

In 2004, members of the RTSRG; Guillem Bernat, Ian Broster, Antoine Colin, and Robert Davis, and the University of York founded a spin-out company called Rapita Systems Ltd. (www.rapitasystems.com) to commercialise the technology and bring it to market. All rights to the technology and prototype tools were transferred to the company by the University of York which became a shareholder in the company.

In 2005, Rapita Systems received £200k of funding from Viking Investments Ltd. and an associated group of Business Angels [6]. Following the initial technology transfer, the pWCET prototype was re-implemented as a commercial quality tool and re-branded as RapiTime. RapiTime has since been extended to support analysis of systems written in C++ as well as the C, and Ada programming languages, and has recently been complemented by a Code Coverage tool (RapiCover) which uses the underpinning RapiTime technology for code instrumentation and analysis. Together, RapiTime and RapiCover are part of the Rapita Verification Suite (RVS).

In 2006, BAE Systems used RapiTime on the Hawk Advanced Jet Trainer project [7]. Here, RapiTime was used to identify opportunities for WCET reduction, thus creating headroom for new functionality to be added to the system, while avoiding the need for a costly hardware upgrade. Using RapiTime, BAE identified that just 1% of hundreds of thousands of lines of code contributed 29% of the overall WCET. Further, by focusing optimisation efforts on this 1% of the code, they were able to reduce the WCET by 23% [8]. Further, RapiTime was quantified as being able to identify timing problems with less than 10% of the effort of previous approaches, potentially saving months of work. As a result Rapita received a BAE chairman's award for Innovation in the category Transferring Best Practice.

Since 2008, Rapita has focused on sales of its RVS product, centred on RapiTime, to customers in the aerospace and automotive markets.

Impact

As described in the previous section, research from the Real-Time Systems Research Group at the University of York was exploited in the development of an innovative Worst-Case Execution time (WCET) analysis technology now called “RapiTime”. This technology was transferred to industry via the formation in 2004 of a successful spin-out company; Rapita Systems Ltd.

RapiTime has been deployed on, and is in continuous use on, a number of major long-term space, aerospace and automotive projects world-wide, examples include: Flight Control Computers [10] and FADECs (Full Authority Digital Engine Control). Alenia Aermacchi (Italy): Flight Control System for the M-346 military transonic trainer. (Since 2010) [9], and various projects for the European Space Agency (ESA).

Since 2008, Rapita has also won significant export orders to China via its distributor Cinawind.

Beneficiaries

RapiTime enables companies in the aerospace and automotive electronics industries to reduce the time and cost required to obtain confidence in the timing correctness of the systems they develop. It provides a cost-effective means of targeting software optimisation, such that new functionality can be added to existing systems without the need for expensive hardware upgrades. Further, RapiTime is portable across a wide range of different microprocessors, meaning that companies can use the same technology across multiple projects without the need for re-training or adoption of multiple solutions.

A major aerospace supplier described the benefits of using RapiTime to identify timing problems during continued development of a Flight Control System as follows: *“The biggest benefit that RapiTime brought to our development process was just how quickly we could get comprehensive timing measurements from our tests. Not only did we reduce our effort requirements for the testing, but we could use our results in ways that were infeasible before. It is now significantly faster for us to identify a timing issue, update the software to resolve the issue, test the updated software and verify that it’s fixed”* - Wayne King, Engineering Fellow – 30th July 2009.

Without RapiTime, the timing measurement and analysis process needed to determine WCETs has to be done manually. This is a painstaking and error prone process that takes considerable time and effort. It also needs to be repeated when changes are made to the application software. Further, the manual process provides no information about the worst-case path, or the contribution of different sections of code to the WCET. This makes code optimisation an ad-hoc, ineffective and inefficient process, as optimising for the worst-case is very different from optimising for the average case.

Alenia Aermacchi engineers working on the M-346 Flight Control System said, *“the main advantage [of using RapiTime] is the possibility to identify software bottlenecks that can be subject to optimisation. Without RapiTime the mandatory code optimisation would have been done without the knowledge of where to concentrate the efforts.”* [9].

Overall, *“Using RVS, customers have cut the worst-case execution time of large scale, legacy applications by up to 50% with only a few days effort, and significantly reduced unnecessary testing and instrumentation overheads”* [10].

Rapita has created and sustained a large number of high technology jobs in York. The success and indeed the existence of the company is a consequence of research undertaken in the Real-Time Systems Research Group at the university of York.

Future Challenges

In the next 5 to 10 years, complex multicore and many-core systems will present an extreme challenge in terms of the difficulty involved in obtaining tight worst-case execution time estimates. The Real-Time Systems Research Group and Rapita Systems are currently collaborating via an EU project PROXIMA [11] on a promising approach to meet this challenge, using probabilistic timing analysis techniques. Rapita also sponsors a number of research students within the RTSRG.

References

- [1] G. Bernat, A. Colin, S. M. Petters, "WCET Analysis of Probabilistic Hard Real-Time Systems" IEEE Real-Time Systems Symposium (RTSS), December 2002, Austin, Texas, USA. DOI: 10.1109/REAL.2002.1181582
- [2] G Bernat, M. J. Newby, A. Burns, "Probabilistic Timing Analysis: an Approach using Copulas" Journal of Embedded Computing, v1-2, pp 179-194, 2005. <http://dl.acm.org/citation.cfm?id=1233760.1233763>
- [3] A. Colin, S. M. Petters "Experimental Evaluation of Code Properties for WCET Analysis" IEEE Real-Time Systems Symposium (RTSS), Cancun, Mexico, December 2003. DOI: 10.1109/REAL.2003.1253266
- [4] A. Colin, G. Bernat, "Scope Tree: a Program Representation for Symbolic WCET Analysis" In Proc. 14th Euromicro Conference on Real-Time Systems (ECRTS), June 2002, Vienna, Austria. DOI: 10.1109/EMRTS.2002.1019185
- [5] G. Bernat, A. Colin, S. M. Petters, "pWCET a Toolset for automatic Worst-Case Execution Time Analysis of Real-Time Embedded Programs" 3rd Int. Workshop on WCET Analysis, at the Euromicro conference on Real-Time Systems, Porto, Portugal, 1 July 2003. (Available as a technical report <https://www.cs.york.ac.uk/ftpdireports/2003/YCS/353/YCS-2003-353.pdf>)
- [6] <http://www.rapitasystems.com/system/files/yabawinter05news.2.pdf>
- [7] http://www.rapitasystems.com/system/files/CaseStudy_BaE_Hawk_2.pdf
- [8] G. Bernat, R.I. Davis, N. Merriam, J. Tuffen, A. Gardner, M. Bennett, D. Armstrong. "Identifying Opportunities for Worst-case Execution Time Reduction in an Avionics System". *Ada User Journal* pp. 189-194, Volume 28, Number 3, Sept 2007.
- [9] http://www.rapitasystems.com/system/files/Aermacchi_case_study_0.pdf
- [10] http://www.rapitasystems.com/system/files/CaseStudy_FlightControl_System_1.pdf

- [11] R.I. Davis, T.Vardanega, J.Andersson, F.Vatrinet, M. Pearce, I. Broster, M. Azkarate-Askasua, F. Wartel, L. Cucu-Grosjean, M. Patte, G. Farrall, F.J. Cazorla" PROXIMA: A Probabilistic Approach to the Timing Behaviour of Mixed-Criticality Systems ". *Ada User Journal, Number 2* , pages 118-122, June 2014.